

## 构件非功能属性描述模型研究

李海文, 南建国, 黄雷, 万明

(空军工程大学航空航天工程学院, 西安 710038)

**摘要:** 非功能属性规约是构件技术研究的难点。属性本身的多样性及与系统其他元素较强的关联性,增加了规约体系建立的难度;同时,在复杂复用环境下属性数据的有效性、准确性难以保障。针对上述问题,将相关构件元素分解,降低系统复杂性,提出规范、统一的非功能属性描述模型;通过属性数据细化过程,提高属性数据准确性;通过继承策略保证数据的有效性。最后给出了多版本属性数据的筛选方法,并提供了具体实例。

**关键词:** 非功能属性; 软件复用; 软件构件化开发; 数据细化

**中图分类号:** TP311 **文献标志码:** A **文章编号:** 1671-637X(2017)01-0076-06

## A Survey on Description Model of Software Component Extra-Functional Properties

LI Hai-wen, NAN Jian-guo, HUANG Lei, WAN Ming

(Institute of Aerospace Engineering, Air Force Engineering University, Xi'an 710038, China)

**Abstract:** The description of Extra-Functional Properties (EFPs) is a challenge in component research. The diversity of the EFPs and their relevance to other element of the system increase the difficulty of description. Meanwhile, it is not easy to guarantee the validity and accuracy of the attribute data. Taking the above issues into consideration, we resolved the element of component to decrease complexity of the system, and introduced a EFPs model, which is normative and uniform. The accuracy of the attribute data was increased through data refining, and the validity of the value was increased through values inherited. Finally, the method for selecting multiple attribute data was given with specific examples.

**Key words:** extra-functional properties; software reuse; Component-Based Software Development (CBSB); data refining

### 0 引言

软件构件化开发(CBSB)是实现软件复用,提高软件开发效率,解决软件危机,增强软件安全性、可靠性的重要手段。传统的构件开发过程大都侧重构件功能属性的规约,对构件的实时性、安全性、存储空间消耗、功耗等非功能属性(Extra-Functional Properties, EFPs)重视不足<sup>[1]</sup>,EFPs又称为服务质量属性(QoS)。随着构件技术在嵌入式软件开发中的应用,仅注重其功能属性规约难以开发出满足用户需求的构件(特别是对于安全关键软件、硬实时性系统软件等)。

针对构件非功能属性规约方法,很多学者做了深

入研究。文献[2]中 Koala 构件模型通过非功能属性配置接口提供 EFPs 属性规约,但该接口只能对静态存储空间这个属性进行描述,无法实现其他 EFPs 属性规约;文献[3]中 Robocop 构件模型通过多样的 EFPs 属性描述模型规约构件非功能属性,如通过资源消耗模型对构件的资源消耗进行评估,利用行为模型对构件操作的调用次序进行定义,但其对软件生命周期支持能力不足;文献[4]中 Palladio 构件模型通过大量注释描述 EFPs 属性,实现构件开发过程中对其性能的评估预测,但描述方式缺乏系统性、规范性。

总结来看,目前对 EFPs 的规约主要存在以下问题:

1) EFPs 缺乏统一、系统的描述体系,现有描述方式大多不够规范,覆盖不够全面,由于 EFPs 的多样化、抽象性,许多属性与软件使用剖面、软件体系结构、底层应用平台乃至整个系统密切相关,增加了对 EFPs 属性规约的难度;

2) EFPs 数据有效性、准确性难以保证,构件是可复用的软件结构,涉及软件整个生命周期,因此,构件 EFPs 数据要伴随构件开发进行有效准确的更新,以便评估预测系统性能,同时,考虑到构件复用环境的复杂性,在不同应用环境中,其 EFPs 数据的有效性、准确性也难以保证。

本文对于上述问题做深入研究后提出一种系统化、体系化的 EFPs 描述模型。该模型支持不同开发阶段、不同应用环境中对 EFPs 数据进行描述,并可实现现在具体构件应用实例中 EFPs 数据的细化,从而保证数据的准确、有效。

### 1 IEDM 模型

本文提出的非功能属性描述模型对 EFPs 属性、属性相关元素及属性数据等进行了高度综合,简称其为 IEDM 模型(Integrative EFPs Description Model)。

IEDM 模型借鉴了 SaveCCM<sup>[5]</sup>对 EFPs 的描述方法(SaveCCM 构件模型中用三元组 < Attribute, Value, Credibility > 表示 EFPs 属性),并在此基础上进行扩展深化,将非功能属性表示为:

$$EFPs = \langle TI, Value \rangle$$

$$Value = \langle Data, Metadata, Validity Condition \rangle$$

其中:TI (Type Identifier) 为非功能属性的类型标识符; Value 为该属性数据集,由属性数据(Data)、属性元数据(Metadata)、属性数据有效条件(Validity Condition)组成,所有的属性描述数据由 Value 存储;Data 为该属性具体应用数据;Metadata 是对 Data 数据的补充说明,描述了 Data 数据的版本、来源、时间戳、准确程度等多样信息;Validity Condition 对 Data 数据的有效条件进行了规定。下面对各描述元素进行详细介绍。

#### 1.1 类型标识符 (TI)

与面向对象编程中类的概念相似, TI 定义了一类属性,并对该类型属性的格式、结构及属性数据的格式等进行了统一规范,构件的应用实例通过继承获得不同属性的描述方法,图 1 所示的类型标识符注册表列举了部分属性类型。图中, TI 标注的类型主要有能源开销(Power Consumption)、最坏情况执行时间(Worst-Case Execution Time, WCET)、静态存储空间消耗(Static Memory Usage)。

TI 注册表有较好的开放性及可扩展性,结合相应的介绍文档,开发人员会对 EFPs 属性的类型有较为系统、全面的认识,并为 EFPs 的进一步规范打下了基础。

为降低 EFPs 数据复杂性与多样性对数据描述的影响,将 EFPs 属性相关的构件元素从构件整体中分离,进行细化归纳,并用统一的规范进行表述,产生了

图 1 的 Attributable 项。EFPs 与 Attributable 项的关系通过 UML 图进行刻画,如图 2 所示,构件的接口(Interface)、连接器(Connector)等与 EFPs 数据相关的元素都可加入 Attributable 项中。

TI Registry	
TI:	Power Consumption
Attributable:	Component
Data Format:	Reference to External Model
Documentation:	...
TI:	WCET
Attributable:	Component, Interface, Operation
Data Format:	Integer
Documentation:	...
TI:	Static Memory Usage
Attributable:	Component
Data Format:	Float
Documentation:	...

图 1 类型标识符注册表

Fig. 1 Attribute type registry

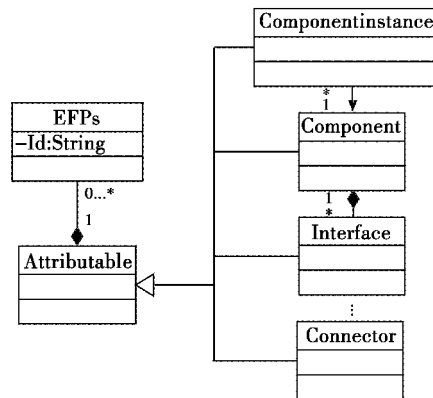


图 2 EFPs 与构件元素的关系

Fig. 2 The relation between EFPs and the elements of component

#### 1.2 属性数据 (Data)

Data 是对属性具体的描述数据,不仅包含传统的数据类型(如 Integer, Float, Array 等),也包含部分复杂数据类型,如图像(Image)、数据分布(Value Distribution)等。

Data 作为 EFPs 属性描述最直接的数据元素,其有效性、准确性对整个模型至关重要,根据开发阶段不同及构件复用环境差异对同一个构件的相同 EFPs 数据进行多版本开发存储。

首先,在构件开发的不同阶段,对相应 EFPs 数据进行及时更新。EFPs 数据在开发过程中可看作是对构件的一种潜在的性能约束,当构件开发完成后,这种约束会内化为软件性能的一部分。模型驱动的软件开发<sup>[6]</sup>(Model-Driven Software Development)优势在于:通过模型分析,尽早发现软件设计中存在的问题,从而提高软件质量,而准确有效的 EFPs 数据便可作为模型分析中的重要参考。IEDM 模型通过开发阶段多版本的

EFPs 属性数据为构件的性能提供了保障。

其次,在构件复用中,不同的应用环境对 EFPs 的要求不同。设计不同版本的 EFPs 属性,并通过筛选机制选择适合复用的 EFPs 版本,可以提高构件复用的准确性。具体方法将在后文详细介绍。

### 1.3 属性元数据 (Metadata)

属性元数据是对不同版本 EFPs 属性数据的有效区分。它对数据的版本 (Version)、来源 (Source)、生成时间 (Time) 等信息进行存储。图 3 定义了部分属性元数据,图中主要列举了版本、来源、时间戳 (EFPs 属性创建及升级时间等)、版本注释信息。

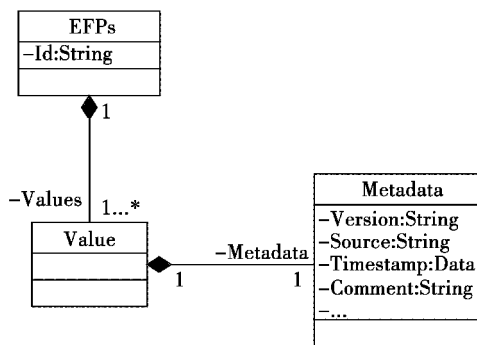


图 3 属性元数据描述

Fig. 3 Description of metadata

### 1.4 数据有效条件 (Validity Condition)

复用性是构件的主要特征之一,对于不同的复用环境,EFPs 数据都应准确描述构件性能,因此构件复用的过程也是 EFPs 数据复用的过程。

数据有效条件明确规定了在何种条件下构件的 EFPs 数据是有效的,这种上下文环境规约包括对底层平台 (Underlying Platform)、使用剖面 (Usage Profile)、其他属性相关性 (Dependencies Towards Other attributes) 等多个方面的约束,具体描述如图 4 所示。

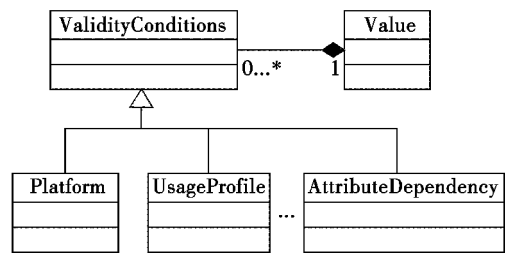


图 4 数据有效条件描述

Fig. 4 Description of validity condition

经过上述介绍,对 IEDM 模型元素有了全面的认识。通过 EFPs 数据相关模型元素的提取,多版本数据的存储,实现了 EFPs 数据的系统描述。IEDM 模型完整描述如图 5 所示。

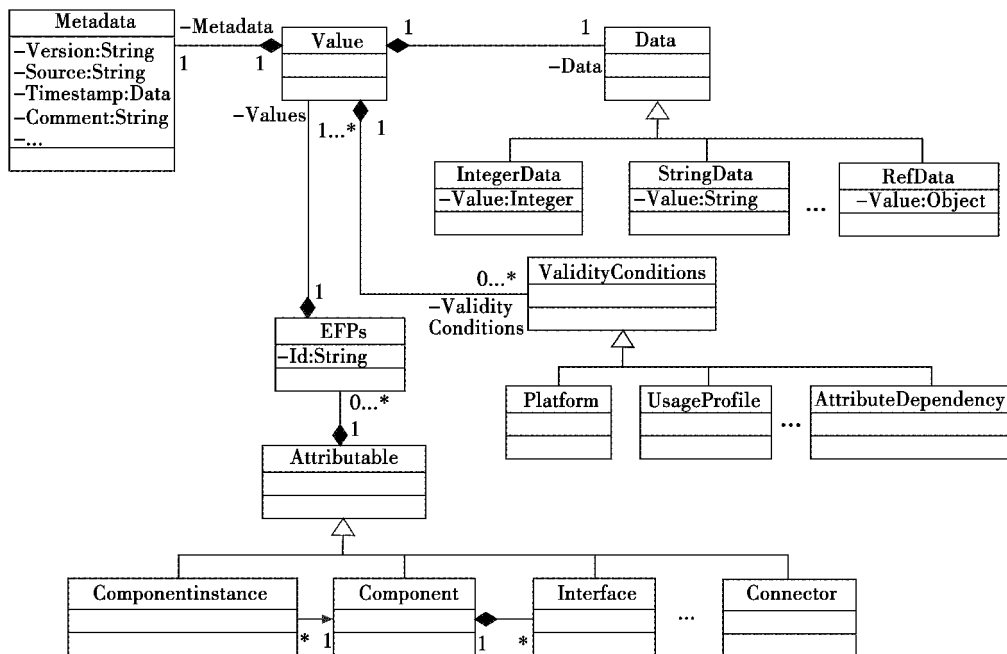


图 5 IEDM 模型完整描述图

Fig. 5 Description of IEDM model

## 2 IEDM 模型数据的细化

针对不同复用环境,设计多版本 EFPs 属性数据,

这些数据的产生来自于对属性元数据的完善性继承,称之为数据的细化 (Refining),这是保证 EFPs 属性数据准确、有效的主要方法。

构件设计的目的是实现软件的复用,对于多样的复用环境,IEDM 模型提供了多版本的 EFPs 属性数据。如图 6 所示的 DB 构件复用实例,构件 DB 分别在不同环境中实现复用,图中给出的 EFPs 数据有 Encryption 数据、QueryLanguage 数据和 WCET 数据。在 DB 所有应用实例中,QueryLanguage 数据完全继承自 DB,WCET 数据则根据不同复用环境,实现了数据的细化(在 DB 中,WCET = 22 ms;而在 DB1'中,由于复用环境的限制,WCET = 12 ms)。

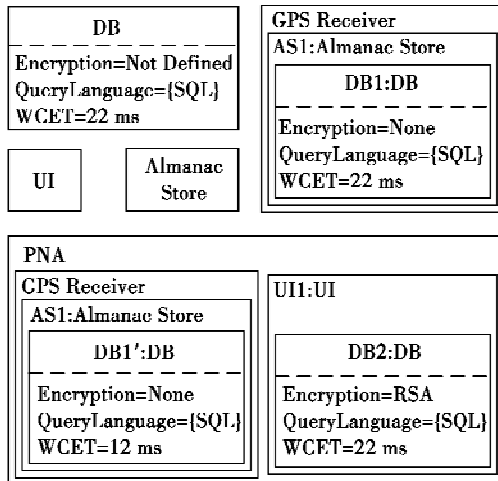


图 6 构件 DB 复用实例

Fig. 6 Reuse example of DB component

上述 EFP 数据的细化过程与面向对象编程中的继承过程相似:DB1, DB1', DB2 被应用于新的环境,它们继承了 DB 中仍然有效的数据元素,对部分数据元素(如 WCET 等)进行了修改,这种修改是在严格的推导规则限制下进行的,从而保证了父子数据的一致性与相容性。

### 2.1 EFPs 数据细化分析

EFPs 数据细化过程可看作某一构件类型(如 DB)与其产生的具体应用实例(如 DB1, DB1', DB2)之间的继承与被继承关系。

通过图 7 表示 IEDM 模型元素与构件类型(Type)、应用实例(Instance)之间的关系及它们在数据细化中的作用。数据的细化由图中 refines 标出;EFPs 数据通过类型标识符进行区分(Identifiable Element);构件类型(Type)也可以继承构件的具体应用实例(Instance),这主要是考虑到多层次的继承问题<sup>[7]</sup>,即该环境下的应用实例可能成为其他环境中的新的类型。Type 与 Instance 可看作父类与子类的关系,Type 的改变会引起 Instance 相应的变化,通过这种继承关系,保证了细化后 EFPs 数据与原数据的一致性。

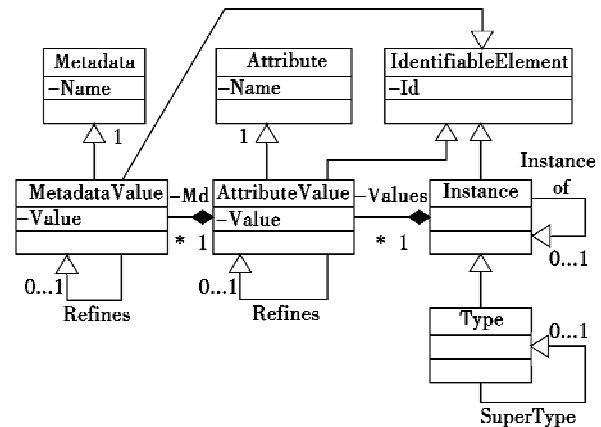


图 7 EFPs 数据细化关系图

Fig. 7 Refinement of EFPs value

### 2.2 EFPs 数据继承策略

在对构件模型元素、EFPs 数据等进行修改时,系统会自动检测对应需要修改的 EFPs 数据;新数据的产生同样需要对原数据进行相应修改,指引这种同步修改的策略成为数据继承策略。

EFPs 数据继承策略可分为属性定义继承策略与属性数据继承策略两类。属性定义继承策略包括 Inherited 策略与 Notinherited 策略两种,分别表示对属性定义的继承与不继承。属性数据继承策略分为 Final 策略、Override 策略、Notinherited 策略 3 种。Final 策略表示细化后的 EFPs 数据继承父属性数据,对于属性数据的修改只能通过父属性变化进行;Override 策略表示细化数据来自系统默认值,但用户可根据相应规则对其进行更改,更改后数据系统会进行校验,保证数据有效性;Notinherited 策略表示该类属性数据从不继承。具体实例如表 1 所示。

表 1 EFPs 数据继承策略实例

Table 1 Example of EFPs attribute inheritance policies

标识符	属性定义继承	属性数据继承	约束
Vendor Name	Notinherited	Notinherited	None
Acquisition Time	Inherited	Override	父属性数据 ≥ 子属性数据
Response Time	Inherited	Override	父属性数据 ≥ 子属性数据
Static Memory	Inherited	Final	None

对于属性数据继承中的 Final 策略与 Override 策略而言,要求必须有相应的属性定义才可继承,因此,这两类继承策略并非相互独立。对于供应商名称(Vendor Name),两类策略都不会继承而是重新定义;而数据获取时间(Acquisition Time)、响应时间(Resp-

onse Time)等则只继承属性定义方式,同时要求子属性数据要在父属性数据规定范围内;静态存储空间(Static Memory)同时继承了两类属性。

总体来看,数据的细化过程主要通过在一定规则下的数据继承实现。通过数据细化实现了构件属性在不同环境中的准确描述;通过继承规则约束,保证了子属性数据的有效性。

### 3 IEDM 模型多版本数据筛选

通过 IEDM 模型规约,建立了较为系统完善的 EFPs 属性描述机制。对于复杂的复用环境及不同开发阶段的需求,定义多版本的 EFPs 数据,通过数据细化策略,保证了 IEDM 模型数据的有效性和准确性。对于不同版本数据,选择数据的过程是复杂的,这里提供了一种属性数据筛选的方法。

在 IEDM 模型中,不同版本数据通过 Metadata 及 Validity Condition 提供的信息进行区分,如版本名称 (Versionname)、时间戳 (Timestamp)、最新版本 (Latest) 等。就筛选过程而言,Metadata 数据与 Validity Condition 数据的作用有很大相似性,不同点在于 Validity Condition 数据可用于对筛选出的信息进行有效性校验,发现其中潜在的冲突。下面对筛选过程进行分析。

用 and 与 or 操作对筛选条件进行简单表述,格式如下<sup>[8]</sup>。

```
Condition1 [ and Condition2 ] or
Condition3 [ and Condition4 ] or
...
```

对于上述筛选条件,先筛选满足 Condition<sub>1</sub> 与 Condition<sub>2</sub> 的 EFPs 数据,若无满足该条件数据则筛选满足 Condition<sub>3</sub> 与 Condition<sub>4</sub> 的数据,而后将满足条件的数据读出。对于如下所示的 EFPs 数据筛选条件:

```
( Platform : X ) and ( Source : Measurement ) or
( Release2.1 ) or
Latest
```

用于筛选图 8 所示的 EFPs 数据(图 8 提供了 EFPs 属性的 Metadata 数据),则最终满足条件的数据有 ①③④⑤标注的数据。其中:⑤所示的 EFPs 属性数据虽然不满足 Condition<sub>1</sub>,但满足 Condition<sub>2</sub>;②⑥标注的数据不满足前两组条件,同时也不是最新的描述数据,无法满足 Condition<sub>3</sub>,因此不会被选出。

在实际的 EFPs 数据筛选过程中,可以通过用户需求及在该任务下其对属性数据的偏好等信息进行数据选择。这里以 OCL 语法<sup>[9]</sup>为基础,给出如下筛选语言语法:

```
SelectCond → AtomCond [ or SelectCond ]
```

```
AtomCond → Mandatory ( AttrCst )
[ and Optional ( AttrCst ) ] |
Optional ( AttrCst )
```

```
AttrCst → an OCL Expression
```

选择条件 (SelectCond) 列举了一系列的选择条件表述 (AtomCond),在选择条件表述中有强制条件 (Mandatory) 与可选择条件 (Optional) 两种,选择过程首先对满足两种条件的数据进行筛选,若没有满足要求数据,则只对满足强制条件的数据进行筛选。AttrCst 为相应条件的描述。

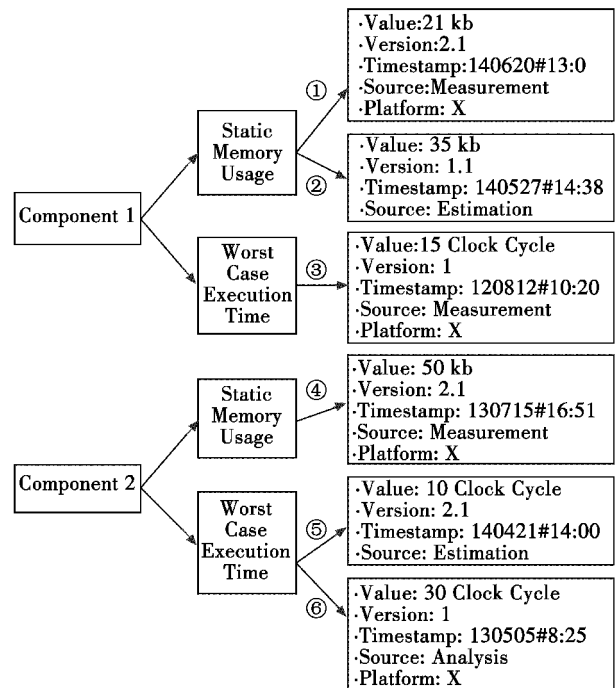


图 8 EFPs 筛选过程举例

Fig. 8 Example of EFPs selection

如果用户要求的强制条件是底层平台为“FreeRTOS”;可选择条件是数据来源为“Measured”;限制条件是 WCET ≤ 20 ms,则可编辑如下选择条件:

```
Global Preference:
Mandatory ( self. Metadata ( " Platform " ) .
StringValue ( ) = ' FreeRTOS ' )
and
Optional ( self. Metadata ( " Measure " ) .
BooleanValue ( ) = ' True ' )
WCET Attribute Preference:
Mandatory ( self. Metadata ( " WCET " ) .
IntValue ( ) < = 20 ms )
```

## 4 结束语

本文针对软件构件化开发中 EFPs 属性描述存在的两类问题展开研究。首先通过 IEDM 模型规范、系

统的描述统一 EFPs 属性描述体系,使属性描述更加全面、科学;而后通过 EFPs 数据细化,保证了属性数据在开发不同阶段及不同复用环境中的准确性、有效性;最后针对 IEDM 模型多版本数据,给出数据筛选方法。该模型对 EFPs 属性的描述还需在开发应用中进一步检验。

### 参考文献

- [1] SENTILLES S, STEPAN P, CARLSON J, et al. Integration of extra-functional properties in component models [C]//The 12th International Symposium on Component Based Software Engineering, 2009:184-197.
- [2] VAN OMMERING R, VAN DER LINDEN F, KRAMER J, et al. The Koala component model for consumer electronics software[J]. Computer, 2000, 33(3):78-85.
- [3] MAASKANT H. A robust component model for consumer electronic products [J]. Philips Research, 2005, 13(3):167-192.
- [4] KOZIOLEK H. Parameter dependencies for reusable performance specifications of software components [D]. Oldenburg: Oldenburg University, 2008.
- [5] AKERHOLM M, CARLSON J, FREDRIKSSON J, et al. The SAVE approach to component-based development of vehicular systems [J]. Journal of Systems and Software, 2007, 80(5):655-667.
- [6] BORDE E, CARLSON J. Toward verified synthesis of ProCom, a component model for real-time embedded systems [J]. IEEE Transactions on Software Engineering, 2011, 20(24):129-138.
- [7] VULGARAKIS A, SURYADEVARA J, CARLSON J, et al. Formal semantics of the ProCom real-time component model [C]//The 35th Euromicro Conference on Software Engineering and Advanced Applications, 2009:478-485.
- [8] 刘美红,金茂忠,晏海华.一种基于 Visual Studio. NET 的 C# 构件模型研究[J]. 计算机工程与应用, 2003, 39(30):62-64, 90.
- [9] 张世琨,王立福,杨芙清.基于 COTS 构件的系统开发[J]. 计算机科学, 2000, 27(1):6-8, 20.

(上接第 66 页)

通过仿真可知,本文所提的制导律能够以期望攻击角命中目标,并具有良好的鲁棒性。

### 5 结束语

本文针对导弹末制导攻击角度约束问题,选取一种快速终端滑模面,设计一种带攻击角约束的有限时间收敛滑模制导律,对有限时间收敛进行分析和证明,并对抖动现象进行削弱分析,通过仿真对比,本文所设计的制导律具有较强的鲁棒性以及较好的制导性能,能够达到所期望的攻击角度约束。

### 参考文献

- [1] KIM M, GRIDER K V. Terminal guidance for impact attitude angle constrained flight trajectories [J]. IEEE Transactions on Aerospace and Electronic Systems, 1973, 9(6):852-859.
- [2] 高峰,唐胜景,师娇,等.一种基于落角约束的偏置比例导引律[J]. 北京理工大学学报, 2014, 34(3):277-282.
- [3] 张旭,雷虎民,曾华,等.带落角约束的自适应比例制导律[J]. 固体火箭技术, 2011, 34(6):687-692.
- [4] 陈琦,王中原,常思江.带有落角约束的间接 Gauss 伪谱最优制导律[J]. 兵工学报, 2015, 36(7):1203-1212.
- [5] 熊少锋,王卫红,刘晓东,等.考虑导弹自动驾驶仪动态特性的带攻击角度约束制导律[J]. 控制与决策, 2015, 30(4):585-592.
- [6] 蔡洪,胡正东,曹渊.具有终端角度约束的导引律综述[J]. 宇航学报, 2010, 31(2):315-323.
- [7] 吴鹏,杨明.带终端攻击角度约束的变结构制导律[J]. 固体火箭技术, 2008, 31(2):116-120.
- [8] 席杰,杨军,朱学平.带落角和导引头视角约束的制导律设计[J]. 电光与控制, 2014, 21(9):26-28.
- [9] 张亚松,任宏光,吴震,等.带落角约束的滑模变结构制导律研究[J]. 电光与控制, 2012, 19(1):66-68.
- [10] 胡正东,郭才发,蔡洪.带落角约束的再入机动弹头的复合导引律[J]. 国防科技大学学报, 2008, 30(3):21-26.
- [11] 刘永善,贾庆忠,刘藻珍.电视制导侵彻炸弹落角约束的变结构导引律[J]. 弹道学报, 2006, 18(2):9-14.
- [12] 常超,林德福,祁载康,等.带落点和落角约束的最优末制导律研究[J]. 北京理工大学学报, 2009, 29(3):233-239.
- [13] 张运喜,孙明玮,陈增强.滑模变结构有限时间收敛制导律[J]. 控制理论与应用, 2012, 29(11):1413-1418.
- [14] 张尧,郭杰,唐胜景,等.机动目标拦截含攻击角约束的新型滑模制导律[J]. 兵工学报, 2015, 36(8):1443-1457.